패킷 마킹을 이용한 해킹경로 역추적 알고리즘

Hacking Path Retracing Algorithm using Packet Marking

원승영
충북대학교 컴퓨터공학과

한승완
한국전자통신연구원 정보보호연구본부
서동일
한국전자통신연구원 정보보호연구본부
김선영
충북대학교 컴퓨터공학과

오창석
충북대학교 전기전자컴퓨터공학부

중심어 : 라우터 ID, 패킷 마킹, 역추적

요 약

현재 DDoS 공격을 근본적으로 차단하여 시스템과 네트워크 자원을 보호하기 위하여 패킷마킹을 이용한 해킹경로 역추적기법들이 연구되고 있다. 기존의 확률적 패킷마킹 역추적 기법은 마킹필드로 IP 식별자필드에 마킹함으로 ICMP의 사용이 불가능하고 경로정보의 암호화와 마크의 크기를 줄이기 위한 hash 함수의 사용으로 암호화된 원래정보의 복원이 불가능하다. 또한 XOR 연산에 의한 결과값의 중복으로 인한 역추적의 문제점을 가지고 있다. 이러한 문제점을 해결하기 위하여 본 논문에서는 라우터 ID를 이용하여 패킷에 마킹하고 마킹된 정보를 추출하여 공격자의 근원지를 정확하고 효율적으로 역추적할 수 있는 알고리즘을 제안하고 구현하였다.

Seung-Young Won
Dept. of Computer Engineering, Chungbuk National University
Seung-Wan Han
Information Security Technology Division, ETRI
Dong-Il Seo
Information Security Technology Division, ETRI
Sun-Young Kim
Dept. of Computer Engineering, Chungbuk National University
Chang-Suk Oh
School of Electrical and Computer Engineering, Chungbuk National University

Abstract

Retracing schemes using packet marking are currently being studied to protect network resources by isolating DDoS attack. One promising solution is the probabilistic packet marking (PPM). However, PPM can't use ICMP by encoding a mark into the IP identification field. Likewise, it can't identify the original source through a hash function used to encode trace information and reduce the mark size. In addition, the retracing problem overlaps with the result from the XOR operation. An algorithm is therefore proposed to pursue the attacker's source efficiently. The source is marked in a packet using a router ID, with marking information abstracted.

Ⅰ. 서론

최근들어 인터넷을 이용한 범죄가 점차 증가하고 있는 추세이다. 그 중에서도 네트워크 기반의 DDoS 공격들은 여전히 치명적인 위협이 되고 있으며, 공격자들에 의해 가장 많이 행해지고 있는 공격기법의 한 종류이다. 공격자들은 자동화된 도구들 예를 들면 TFN, Trinoo, Stacheldraht[1]등을 이용하여 기존의 공격보다 위협적이고 성공적인 공격을 가하고 있다. 따라서 공격자들로부터 DDoS 공격이 이루어질 때 근본적으로 공격자를 검출하고 근절하여 시스템과 네트워크를 보호하는 것이 매우 중요하게 되었다. 하지만 이러한 DDoS를 차단하기 위한 다양한 기법들의 연구가 진행되고 있으나 아직은 정확성과 효율성 면에서 부족한 현실이다. 기존의 역추적 기법은 시스템의 로그 파일이나 중요 파일 변경 여부를 분석하여 추적했다. 그러나 공격자에 의해 로그 파일이나 중요파일이 지워진다면 역추적을 할 수 없다는 문제점을 가지고 있다. 이러한 기존의 역추적 기법을 보완하려는 여러 가지 방법들이 제안 및 연구되고 있으나 현실적으로 아직도 체계적이지 못한 실정이다.

※ 본 연구는 한국전자통신연구원 연구과제로 수행되었습니다.

본 논문에서는 DDoS공격이 이루어질 때 패킷 마킹을 통하여 해킹경로를 역추적하는 알고리즘을 제안하고 구현하여, 기존의 역추적 기법보다 좀더 효율적이고, 정확한 역추적을 할 수 있었다. 본 연구 결과의 활용을 통해 DDoS 공격으로 인한 피해를 줄일 수 있고, 역추적 방법에 있어서 좀더 개선된 기법이 될 수 있을 것으로 기대된다.

II. 패킷 마킹 기술

1. 개념

일반적으로 패킷 마킹은 기존의 콘텐츠 보호를 위한 워터마킹과는 달리 패킷에 임의의 마크를 삽입하여 네트워크 트래픽 제어 및 역추적을 위해 마킹하는 것을 말한다. 이 기법은 라우터가 네트워크에서 패킷이 전송되는 동안 패킷의 경로 정보를 확률적으로 패킷에 마킹을 하게된다. 마킹된 패킷을 받은 피해호스트는 이를 이용하여 네트워크상에서 공격자의 근원지를 찾아가게 되는 것이 확률적 패킹 마킹이다. 그림 1은 일반적인 확률적 패킷 마킹 기법의 구조를 나타내고 있다.

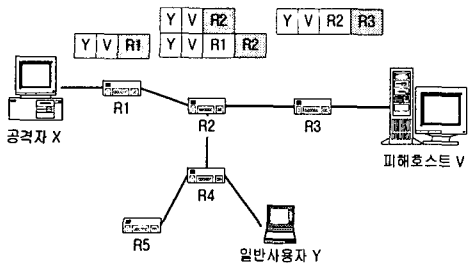


그림 1. 확률적 패킷 마킹 구조

2. 확률적 패킷 마킹 기술

확률적 패킷 마킹 기술의 기본은 패킷이 전송되는 동안 라우터에서 패킷에 확률적으로 경로정보를 마킹하는 것이다. 이렇게 마킹된 패킷을 받은 피해호스트는 마킹된 패킷을 이용하여 공격자의 근원지를 네트워크상에서 찾아가게 되는 기법이다. 그 중에서 Node append 기법은 패킷 마킹 기법 중에서 가장 단순한 알고리즘으로 패킷이 경유하는 라우터의 정보를 패킷에 순차적으로 추가시키는 기법이다. 그러나 라우터들의 오버헤드와 경로의 전체길이를 모르기 때문에 패킷의 공간 확보에 대한 불확실성, MTU 같은 서비스에 불필요한 단편화를 초래할 수 있다.

Node sampling 기법은 라우터의 트래픽 증가와 패킷 공간 확보 문제를 해결하기 위하여 경로를 샘플링하여 기록하는 기법이다. 각 라우터는 확률 p를 이용하여 IP 헤더에 경로정보를 마킹하게 된다. 만약 모든 라우터에 동일한 마킹확률 p를 적용하게 되면, d홉 떨어진 라우터로부터 마킹된 패킷을 받을 확률은 $p(1-p)^{d-1}$이다[3],[4],[5].

Edge sampling기법은 위에 제시된 두 기법의 문제들을 해결하기 위해 공격경로의 edge를 암호화하는 것이다. 거리필드와 라우터의 IP 주소를 표현하기 위하여 start주소 필드와 end주소 필드가 필요하다. 라우터가 패킷에 마킹을 결정하게 되면, start 필드에 라우터 자신의 IP 주소를 기록하고 거리 필드에 0을 기록한다. 거리가 d홉 떨어진 라우터로부터 마킹된 패킷을 받을 확률은 $p(1-p)^{d-1}$이다. 그리고 라우터에서 마킹된 패킷이 도착할 확률은 적어도 $dp(1-p)^{d-1}$이다.

Advanced Marking Scheme기법은 그림 2의 인코딩 구조에 나타나듯이 IP 식별자 필드 16비트를 마킹 필드로 사용하며 라우터까지의 거리를 나타내는 5비트의 거리 필드와 11비트의 edge 필드로 구분한다. 5비트의 거리필드는 32홉을 표시할 수 있으므로 인터넷 경로들을 충분히 나타낼 수 있다. 패킷의 IP 식별자 필드에 32비트의 라우터 주소를 hash[4]함수를 이용하여 11비트로 암호화[6]한 후 마킹하게 된다. 그 후 각 라우터를 경유할 때마다 XOR 연산을 통해 라우터의 정보를 암호화하여 마킹하게 되고, 공격경로를 재설정하게 된다.

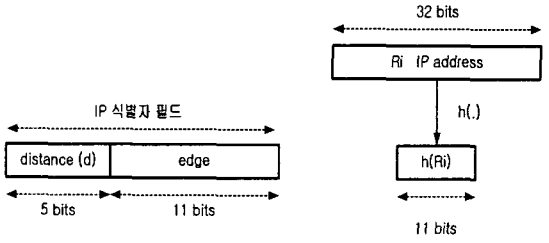


그림 2. Advanced Marking Scheme 인코딩 구조

거리필드는 각 라우터가 확률 p로 패킷을 마킹할 때 마킹된 라우터와의 거리필드가 된다. 즉 자신이 마킹되면 그 라우터의 거리값은 0 이 된다. edge 필드에는 hash 함수를 이용하여 32비트의 IP 주소를 11비트로 암호화하여 저장한다. 확률 p로 마킹이 결정된 한 라우터 R_i 에서는 $h(R_i)$를 edge 필드에 마킹하고 거리필드에 0을 마킹하게 된다. 이미 거리 필드의 값이 0이면 이전 라우터가 마킹을 한 것임으로 edge 필드에 $h'(R_i)$와 XOR 연산한 값을 마킹한다. 지나

온 상위 경로를 알아내는 재설정 절차는 $z=x\oplus h'(y)$로서 y는 upstream 상의 한 라우터이며 x는 $h(R_i)$의 hash값을 나타낸다. 재설정은 a⊕b⊕a = b에 의해 마킹된 edge 필드를 피해호스트로부터 경로를 따라 XOR 연산을 이용하면 최종적으로 공격자 위치를 역추적 할 수 있음을 보여준다. 그림 3은 XOR 연산을 이용한 마킹 절차와 경로 재설정 절차를 나타낸다.

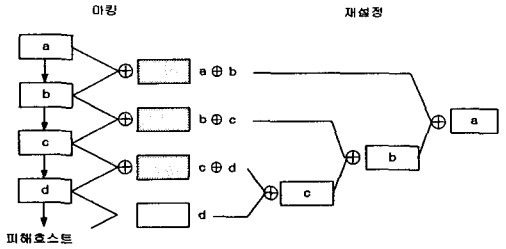


그림 3. XOR 연산을 이용한 마킹과 재설정

3. 패킷 마킹의 문제점

지금까지 현실적인 제안사항과 다양해지는 공격들에 의해 공격자의 근원지를 정확하게 역추적할 수 있는 알고리즘은 아직 개발되지 못하는 실정이다. 기존의 패킷 마킹 기법은 단지 경유하는 라우터의 주소를 패킷에 순차적으로 마킹하거나, 혹은 IP 식별자 필드를 이용하여 라우터에서 확률적으로 마킹하는 기법이었다. 이러한 기법들은 패킷의 공간 확보에 대한 불확실성과 MTU 같은 서비스에 불필요한 단편화의 문제점과 IP 식별자 필드를 마킹필드로 사용함으로써 ping과 traceroute와 같은 ICMP를 사용하지 못하는 문제를 가지며 경로정보를 암호화하기 위하여 단방향 함수인 hash 함수를 사용하여 암호화되기 이전의 원래 정보는 알아낼 수 없는 문제점을 가지고 있다. 따라서 이러한 문제점들을 해결하고 좀 더 정확하고 효율적인 역추적을 하기 위하여 본 논문에서는 패킷 마킹 기법을 이용한 해킹경로 역추적 알고리즘을 제안하고 구현하였다.

III. 패킷 마킹을 이용한 해킹경로 역추적 알고리즘

1. 패킷 마킹을 이용한 해킹경로 역추적 알고리즘 제안

최근 역추적을 위한 패킷 마킹 기법들이 제안되어 왔으나 위에서 제시한 문제점으로 인해 정확하고 효율적인 역추적과 패킷 마킹이 필요하다. 본 논문의 패킷 마킹을 이용한 해킹경로 역추적 알고리즘을 제안하고 구현하기 위해서 다음과 같은 요구사항이 필요하다.

- 공격자는 피해호스트에게 다수의 패킷을 전송할 수 있다.
- 다수의 공격자가 존재할 수 있다.
- 공격자는 역추적 당하고 있다고 인지할 수도 있다.
- 라우터는 신뢰할 수 있다.
- 라우터들과 피해호스트는 서로 통신할 수 있다.
- 라우터는 IP 주소에 대응되는 고유의 ID를 가지고 있다.
- 피해호스트는 라우팅 맵을 가지고 있다.

그림 4는 패킷 마킹을 이용한 해킹경로 알고리즘의 흐름도이다.

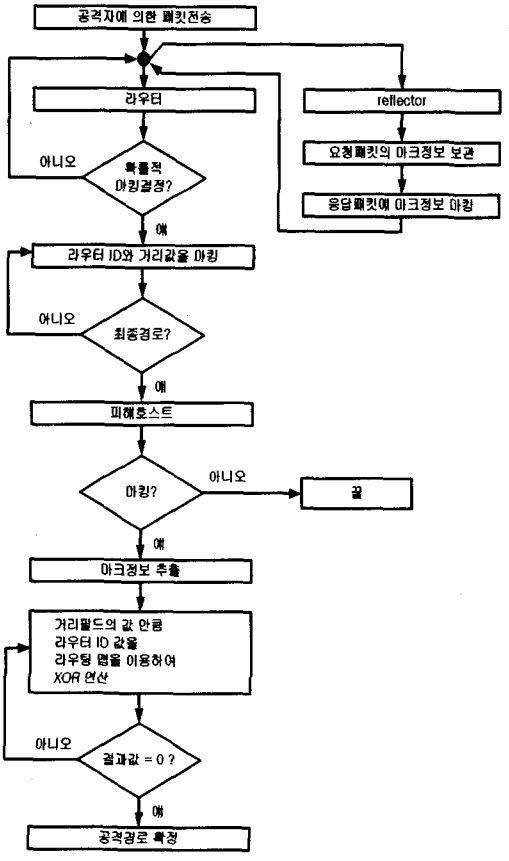


그림 4. 패킷 마킹 알고리즘 흐름도

패킷 마킹을 이용한 해킹경로 역추적 알고리즘의 개요는 다음과 같다.

■ 공격자에 의해 DDoS 공격을 위한 패킷이 전송된다.
■ 패킷이 경유하는 라우터에서는 확률적으로 패킷에 라우터의 ID와 거리값을 마킹하다.
  ◦ 최초의 마킹에서는 거리필드에 0과 라우터 ID 필드에 현재 라우터 ID값을 마킹한다.
  ◦ 다음 라우터에서는 확률적 마킹이 아닌 거리필드의 값을 1 증가시키고 라우터 ID 값을 XOR 연산을 이용하여 마킹한다.
■ 마킹된 패킷을 받은 피해 호스트는 패킷의 마크정보와 라우팅 맵을 이용하여 공격자의 근원지를 찾아간다.
  ◦ 마킹된 패킷에서 마크정보를 추출하여 거리필드의 값만큼 역으로 XOR 연산을 한다.
  ◦ 연산된 결과값이 0이면 패킷이 경유한 경로이다.

2. 패킷 마킹

공격자에 의해 DDoS 공격에 사용된 패킷이 전송될 경우 라우터들은 패킷에 확률적으로 마킹하게 된다. 그림 5는 IP 헤더에서의 마킹 위치를 보여주고 있다[7][9].

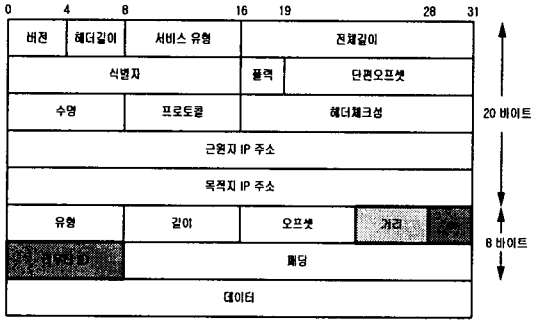


그림 5. IP 헤더에서의 마킹 위치

IP 헤더의 옵션필드를 사용하기 위해 유형필드, 길이필드, 오프셋필드가 필요하다. 그림 6은 IP 옵션 필드를 사용하기 위한 유형필드를 나타내고 있다[8].

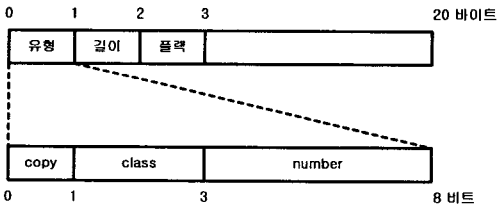


그림 6. 유형필드의 구조

유형필드는 IP 헤더에서 옵션필드의 사용여부를 구분하는 copy 필드, 옵션필드의 컨트롤과 디버깅기능을 나타내는 class 필드, number 필드로 나누어진다. number 필드는 현재 RFC에 0에서 24까지 정의되어 사용되고 있다. 본 논문에서 제안한 알고리즘에서는 네트워크의 과부하를 줄이기 위해 copy 필드를 0으로, class 필드는 디버깅을 나타내는 2로, number 필드는 현재 사용되지 않는 29번으로 마킹하였다. 옵션필드의 두 번째 바이트인 길이필드는 유형필드부터 패딩까지의 바이트 수를 나타내며, 오프셋필드는 IP 옵션필드의 데이터 시작부분을 바이트로 나타낸다. 그 이후의 2바이트가 마킹에 사용될 마킹필드이다. 그림 7은 마킹필드의 구조를 나타낸다.

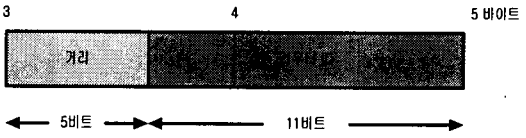


그림 7. 마킹필드의 구조

마킹 필드로 사용하는 2바이트 중 5비트는 피해호스트와의 거리를 표현하는 거리필드로 사용되며, 나머지 11비트는 라우터의 ID를 마킹하기 위한 라우터 ID 필드로 사용된다. 거리필드는 라우터에서 패킷에 마킹을 결정하게 되면 0을 마킹하고, 라우터 ID필드에는 현재의 라우터 ID를 마킹하게 된다. 그렇지 않고 거리필드의 값이 null 값이 아니라면, 이전의 라우터에서 이미 마킹되어진 패킷이므로 거리필드의 값을 1 증가시키고 이전의 라우터 ID 필드 값과 현재 라우터 ID 값을 XOR 연산하여 라우터 ID 필드에 마킹하게 된다. 표 1은 라우터 ID를 나타내고 있다.

표 1. 라우터 ID

라우터 IP주소	라우터 ID	라우터 IP주소	라우터 ID
R1	000 0001 0000	R11	000 1011 0000
R2	000 0000 0010	R12	000 0000 1100
R3	000 0011 0000	R13	000 1101 0000
R4	000 0000 0100	R14	000 0000 1110
R5	000 0101 0000	R15	000 1111 0000
R6	000 0000 0110	R16	001 0000 0000
R7	000 0111 0000	R17	001 0001 0000
R8	000 0000 1000	…	…
R9	000 1001 0000	R127	111 1111 0000
R10	000 0000 1010	R128	000 0000 0000

기존의 패킷 마킹 기법에서는 마크정보의 크기를 줄이기 위해 라우터의 주소를 hash 함수로 암호화하여 사용하였다. 하지만 hash 함수는 단방향 함수로써 암호화된 데이터에서 다시 암호화 이전의 데이터로 복원이 불가능하다. 본 논문에서는 이러한 암호화 문제를 해결하기 위해 라우터 IP주소와 유일하게 대응되는 라우터 ID를 사용하여 해결하였다. 11비트의 라우터 ID는 128개의 라우터를 표현할 수 있으며, XOR 연산 결과 값의 중복을 피하기 위하여 위의 표 1에서와 같은 수 체계를 사용하였다.

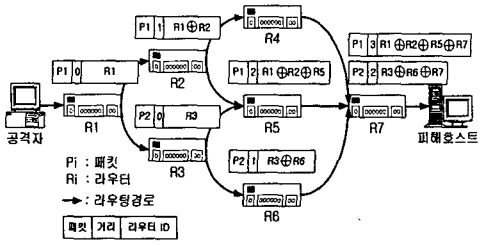


그림 8. 라우터에서 패킷에 마킹되는 과정

그림 8에서 패킷 P1의 마킹과정을 살펴보면, 라우터 R1에서 처음으로 패킷에 마킹을 시작하였다. 거리필드에는 0을 마킹하고, 라우터 ID에는 R1의 라우터 ID 값인 00000010000을 마킹한다. P1을 받은 R2에서는 거리필드의 값이 null이 아니기 때문에 거리필드의 값을 1 증가시키고 R1의 ID값과 R2의 ID 값을 XOR 연산하여 마킹하게 된다. 즉, R2에서는 R1⊕R2의 값을 라우터 ID필드에 마킹하게 된다. 다음은 최종 라우터인 R7까지의 마킹되는 과정을 나타낸다.

- R1(00000010000)⊕R2(00000000010) = 00000010010
- R1⊕R2(00000010010)⊕R5(00001010000)
  = 00001000010
- R1⊕R2⊕R5(00001000010)⊕R7(00001110000)
  = 00000110010

위와 같이 피해호스트에서 받는 거리필드의 값은 3이고 라우터 ID 필드의 값은 00000110010가 된다. 그림 9, 10은 마킹 알고리즘과 reflecting[10] 알고리즘을 도시하였다.

```
Marking procedure at router Ri;
  let p be a marking probability
  let mark_fields be a fields for mark
  (distance, router ID)
  let MARK be a mark flag (defined 0x5d)
  let ID be a array for router IDs
  for each packet P
      let x be a random number from [1..100)
      if x <= p then
          if mark_fields[0] is not MARK then
              if ip.header_length <= 5
                  insert mark_fields in P
                  P.distance <- 0
                  P.ID <- ID(Ri)
          else
              P.distance <- P.distance + 1
              P.ID <- P.ID ⊕ ID(Ri)
```

그림 9. 마킹 알고리즘

```
Storing procedure at Reflector Rf;
  let T be a table for storing
  let entry in T be tuples( mark_info, flag )
  let mark_fields be a fields for mark
  (distance, router ID)
  for each incoming request packet P
    if ip.header_length is marked length
        and mark_fields[0] is MARK
        entry(mark_info) <- P.mark_fields
        entry(flag) <- true
Reflecting procedure at Reflector Rf;
 for each outgoing reply packet P
     if entry(flag) is true then
     insert mark_fields in P
    write entry(mark_info) into P.mark_fields
```

그림 10. Reflecting 알고리즘

3. 공격경로의 재설정

마킹되어진 패킷이 피해호스트에 도착하게 되면 피해호스트에서는 마크 정보 값과 라우팅 맵을 이용하여 공격자의 근원지를 역추적하게 된다. 그림 11은 피해호스트에서 가지고 있는 라우팅 맵을 나타내고 있다.

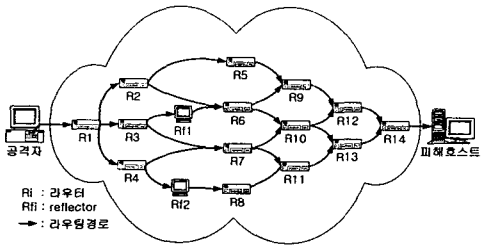


그림 11. 라우팅 맵

공격경로의 재설정 알고리즘은 라우터에서의 마킹 알고리즘과 같이 피해호스트에서 XOR 연산에 의해 이루어진다. 예를 들어, 공격자로부터 전송되어진 패킷이 R1->R3->R7->R10->R13->R14을 경유하여 피해호스트에 도착했다고 가정한다면, 그에 대한 거리필드의 값은 5일 것이며 라우터 ID 필드의 값은 R1⊕R3⊕R7⊕R10⊕R13⊕R14의 값이 마킹되며, 00010000100일 것이다. 이러한 마크의 정보들과 라우팅 맵을 이용하여 최초로 마킹된 라우터 R1까지의 정확한 공격경로를 역추적 할 수 있다. 다음은 그림 11의 라우팅 맵에 의하여 공격자가 피해호스트까지의 공격경로로 이용할 수 있는 경우의 수는 다음의 14가지이다.

■ 공격자-> R1->R2->R5-> R9->R12->R14 -> 피해호스트
■ 공격자-> R1->R2->R6-> R9->R12->R14 -> 피해호스트
■ 공격자-> R1->R2->R6->R10->R12->R14 -> 피해호스트
■ 공격자-> R1->R2->R6->R10->R13->R14 -> 피해호스트
■ 공격자-> R1->R3->R6-> R9->R12->R14 -> 피해호스트
■ 공격자-> R1->R3->R6->R10->R12->R14 -> 피해호스트
■ 공격자-> R1->R3->R6->R10->R13->R14 -> 피해호스트
■ 공격자-> R1->R3->R7->R10->R12->R14 -> 피해호스트
■ 공격자-> R1->R3->R7->R10->R13->R14 -> 피해호스트
■ 공격자-> R1->R3->R7->R11->R13->R14 -> 피해호스트
■ 공격자-> R1->R4->R7->R10->R12->R14 -> 피해호스트
■ 공격자-> R1->R4->R7->R10->R13->R14 -> 피해호스트
■ 공격자-> R1->R4->R7->R11->R13->R14 -> 피해호스트
■ 공격자-> R1->R4->R8->R11->R13->R14 -> 피해호스트

거리필드의 값이 5이고 00010000100의 라우터 ID 필드 값을 위의 14가지 경우를 차례대로 XOR 연산을 사용하여 경로를 역으로 5번 연산하면 최종 결과 값인 00000000000 일 때의 경로가 공격패킷이 경유한 라우팅 경로이다. 다음은 XOR 연산을 이용한 경로 재설정 과정을 보여준다.

■ 피해호스트에서 받은 패킷의 라우터 IDfield의 값 : 00010000100
■ 피해호스트에서 받은 패킷의 거리필드의 값 : 5
- 00010000100 ⊕ R14(00000001110) = 00010001010
- 00010001010 ⊕ R13(00011010000) = 00001011010
- 00001011010 ⊕ R10(00000001010) = 00001010000
- 00001010000 ⊕ R7(00001110000) = 00000100000
- 00000100000 ⊕ R3(00000110000) = 00000010000
- 00000010000 ⊕ R1(00000010000) = 00000000000

XOR를 이용하여 연산한 결과 00000000000의 값이 되었다. 즉, 피해 호스트에서 받은 패킷의 라우터 ID 필드의 값인 00010000100은 라우터 R1->R3->R7->R10->R13->R14의 경로를 통해 피해호스트까지 전달될 것임을 알 수 있다. 이렇듯 거리필드의 값과 라우터 ID를 이용하여 정확한 공격경로를 역추적할 수 있다. 그림 12는 패킷 마킹을 이용한 경로 재설정 알고리즘이다.

```
Reconstruction procedure at Victim V;
 let ip be a IP header of incoming packet
 let router_path be a one of the routing path
 in router map
 let RID be a one of the router ID in router_path
  for each packet P
   if ip.header_length is marked length
          and mark_fields[0] is MARK
      let m be a mark_fields in P
      for each routing_path
         for 0 to m.distance
            Get RID from router_path
            m.ID <- m.ID ⊕ RID
         if m.ID is 0
  extract path(Ri..Rj) by path in router_path
```

그림 12. 패킷 마킹을 이용한 경로 재설정 알고리즘

4. 패킷 마킹을 이용한 해킹경로 역추적 알고리즘 구현

본 논문에서 제안한 패킷 마킹을 이용한 해킹경로 역추적 알고리즘을 구현하기 위한 모듈은 크게 Sender, Passer, Receiver 모듈로 나누어진다.

4.1. Sender 모듈

Sender는 공격패킷을 생성하여 전송할 수 있는 공격자

역할의 모듈이다. 임의의 패킷을 생성하여 피해호스트에게 전송하는 역할을 수행한다. 패킷이 생성되어 전송되는 과정을 살펴보면, sender에서는 일반적인 IP헤더 20바이트 + TCP 헤더 20바이트 + 데이터 형태로 패킷을 생성한다. IP 헤더의 근원지 IP 주소에는 현재 sender가 설치되어 있는 호스트의 IP 주소를 설정하고 목적지 IP 주소에는 receiver가 설치되어 있는 호스트의 IP 주소를 설정한다. 그림 13에 Sender 모듈의 처리 흐름도를 도시하였다.

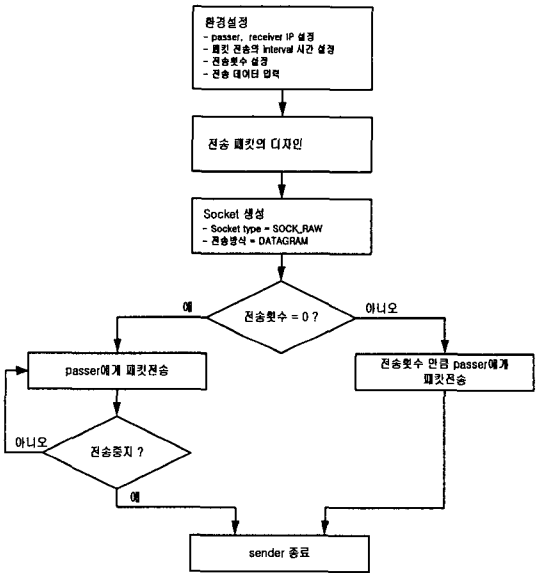


그림 13. Sender 모듈의 처리 흐름도

4.2. Passer 모듈

Passer는 라우팅 기능과 패킷 마킹 기능을 담당하는 모듈이다. 패킷이 라우팅 경로를 통해 전송될 때 라우터 ID를 이용한 패킷 마킹이 이루어지게 된다. 이러한 처리과정을 살펴보면, sender로부터 패킷을 수신하기 위하여 passer의 환경을 설정하게 된다. 여기에서는 패킷수집을 위한 인터페이스 설정, 필터링을 위한 필터 설정 후 receiver의 IP 주소와 라우터에서 마킹할 확률을 설정하게 된다. 이러한 환경설정이 끝나면, 패킷을 수신하게 되는데, 이때 수신된 패킷의 도착시간, 프로토콜 근원지 IP 주소, 목적지 IP 주소를 출력한다. 마킹이 결정되면 마킹을 위한 공간확보를 위하여 TCP 헤더부터 마킹할 공간만큼 데이터가 뒤로 이동되게 된다. 만약, 최초의 패킷 마킹이면 거리필드에 0, 라우터 ID 필드에 현재 라우터의 ID를 마킹하게 된다. 그렇지 않고 이미 마킹된 패킷이라면 거리필드의 값을 1 증가시키고, 라우터 ID필드에 현재 라우터 ID와 이전의 라우터 ID 필드의 값을 XOR 연산한 결과값을 마킹하게 된다. 그림 14는 passer 모듈의 처리과정을 보여주는 흐름도이다.

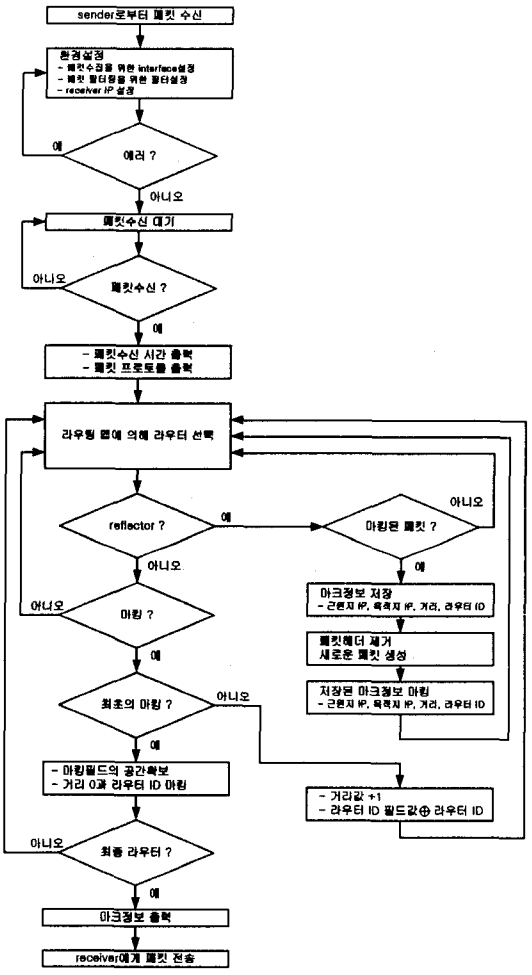


그림 14. Passer 모듈의 처리 흐름도

4.3. Receiver 모듈

Receiver는 공격경로를 재설정하는 기능을 담당하는 피해호스트 역할의 모듈이다. receiver에서의 처리과정을 살펴보면 수신된 패킷의 도착시간, 프로토콜, 근원지 IP 주소, 목적지 IP 주소를 출력하고 마킹 여부를 판단한다. 마킹된 패킷이면 마크정보를 추출하여 거리필드의 값만큼 라우터 ID 필드의 값을 라우팅 맵을 이용하여 역으로 XOR 연산을 하여 결과값이 0이면 패킷이 지나온 공격경로와 일치함으로 그 경로를 출력한다. 그림 15는 receiver 모듈의 처리 흐름도를 도시하였다.

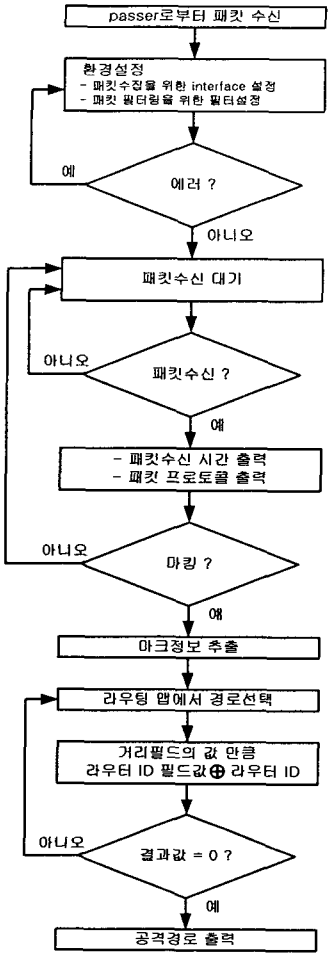


그림 15. Receiver 모듈의 처리 흐름도

IV. 실험 및 고찰

본 논문에서 제안하고 구현한 실험 시스템의 구성환경은 PC 3대에서 운영체제는 윈도우즈 XP 환경, 구현언어는 Visual C++을 사용하여 구현하였다.

1. 실험 시스템의 구성

패킷 마킹을 이용한 해킹경로 역추적 알고리즘 검증을 위한 실험 환경 구성도는 그림 16과 같다. 윈도우즈 시스템에 Sender, Passer, Receiver 모듈을 각각 설치하였다. 실험 환경 구성은 그림 16과 같이 도시하였다.

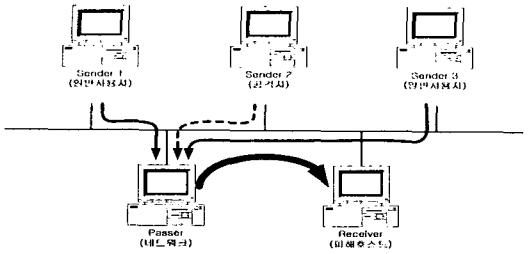


그림 16. 실험 환경 구성도

2. 실험 내용 및 결과

Sender에서는 공격자 역할로 전송간격은 실험을 위해 5ms로 설정하였고 전송횟수는 DDoS공격을 위해 1,000,000개로 설정하였으며, 패킷의 데이터부분인 payload에는 공격자라는 스트링을 삽입하였다. 그림 17은 sender 모듈의 설정이 끝나고 패킷을 전송하는 그림이다.

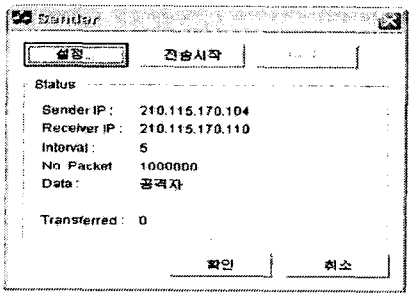


그림 17. 패킷 전송

Passer는 sender로부터 전송된 패킷을 수신하여 필터를 설정하고 마킹할 확률을 설정 후 라우팅과 마킹을 시작한다. 그림 18,19는 passer에 의해 마킹확률의 설정과 설정에 의해 처리된 패킷정보를 도시하였다.

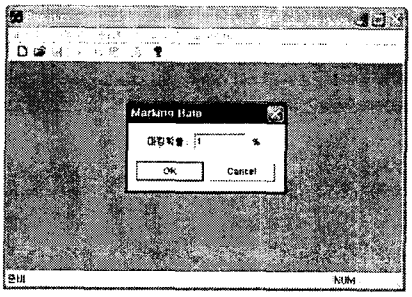


그림 18. Passer에서의 마킹확률 설정

No.	Routing Info	Distance	First Ma...	Data	Payload
9253	01->02->06->10->13->14->Victim				공격자
9254	01->04->R12->08->11->13->14->Victim				일반사용자
9255	01->04->R12->08->11->13->14->Victim				공격자
9256	01->04->R12->08->11->13->14->Victim	4	4	0x2062	공격자
9257	01->04->07->11->13->14->Victim	3	7	0x181e	공격자
9258	01->02->06->10->13->14->Victim				일반사용자
9259	01->02->05->09->12->14->Victim				공격자
9260	01->02->06->10->12->14->Victim	5	1	0x281c	공격자
9261	01->02->05->09->12->14->Victim	3	5	0x18c2	공격자
9262	01->03->R11->06->10->12->14->Victim				공격자
9263	01->02->06->09->12->14->Victim				일반사용자
9264	01->02->06->10->13->14->Victim				공격자
9265	01->02->06->09->12->14->Victim				공격자
9266	01->02->05->09->12->14->Victim				공격자
9267	01->02->06->10->12->14->Victim				공격자

그림 19. Passer에 의해 처리된 패킷 정보

그림 19는 패킷들이 수신된 번호, 라우팅 정보, 거리 필드의 값과 최초로 패킷에 마킹한 라우터의 ID, 라우터 ID필드의 값을 보여주고 있다. 이러한 값들은 receiver에 의해 정확하게 공격경로를 역추적 할 수 있는지 확인하기 위한 자료로 사용될 것이다. 그림 20은 passer에 의하여 1%의 마킹 확률로 마킹된 패킷이 receiver에게 전송되면 receiver는 수신된 패킷의 마킹유무를 확인하여 마킹된 패킷에 대하여 마크정보를 추출한다. 추출된 마크정보와 라우팅 맵을 이용하여 공격경로를 역추적할 수 있다.

No.	Time	Source IP	Destinat...	Reconstructin Path	D	Payload
009253	01/11-17:20:37	210.115...	210.115....			공격자
009254	01/11-17:20:37	210.115...	210.115....			일반사용자
009255	01/11-17:20:37	210.115...	210.115....			공격자
009256	01/11-17:20:37	10.10.1....	210.115....	Victim->14->13->11->8->4	4	공격자
009257	01/11-17:20:37	210.115...	210.115....	Victim->14->13->11->7	3	공격자
009258	01/11-17:20:37	210.115...	210.115....			일반사용자
009259	01/11-17:20:37	210.115...	210.115....			공격자
009260	01/11-17:20:37	210.115...	210.115....	Victim->14->12->10->6->2->1	5	공격자
009261	01/11-17:20:37	210.115...	210.115....	Victim->14->12->9->5	3	공격자
009262	01/11-17:20:37	210.115...	210.115....			공격자
009263	01/11-17:20:37	210.115...	210.115....			일반사용자
009264	01/11-17:20:37	210.115...	210.115....			공격자
009265	01/11-17:20:37	210.115...	210.115....			공격자
009266	01/11-17:20:37	210.115...	210.115....			공격자
009267	01/11-17:20:37	210.115...	210.115....			공격자

그림 20. Receiver에 의해 역추적된 경로

receiver로 수신된 9261번째 패킷의 역추적된 경로를 살펴보면 피해호스트로부터 최초로 마킹을 시작한 라우터까지의 역추적된 경로는 R14->R12->R9->R5이다. receiver에서 추출된 패킷의 마크정보는 0x18c2(00011 00011000010)이며 이 값은 앞에서 언급된 라우팅 맵을 이용한 XOR 연산의 결과값이다. 다음의 연산을 통하여 정확한 역추적의 결과임을 확인할 수 있다.

■ Receiver에서 추출된 패킷의 라우터 ID 필드의 값 : 00011000010

■ Receiver에서 추출된 패킷의 거리필드의 값 3 : (00011)

- 00011000010 ⊕ R14(00000001110) = 00011001100
- 00011001100 ⊕ R12(00000001100) = 00011000000
- 00011000000 ⊕ R9(00010010000) = 00001010000
- 00001010000 ⊕ R5(00001010000) = 00000000000

따라서 본 논문에서 패킷 마킹을 이용한 해킹경로 역추적 알고리즘을 이용하여 구현한 소프트웨어로 실험한 결과 DDoS 공격시 해킹경로 역추적이 기존보다 정확하고 효율적임을 알 수 있다.

V. 결론

패킷 마킹을 이용한 해킹경로 역추적 알고리즘은 DDoS 공격에 대비하여 근본적인 DDoS 공격의 근절과 시스템 및 네트워크의 자원을 보호하기 위한 목적으로 제안하고 구현하였다. 공격자가 DDoS 공격을 위해 전송한 패킷이 라우팅되는 동안 라우터에서 패킷 헤더의 옵션필드에 라우팅 경로정보를 마킹한다. 이렇게 마킹된 패킷을 받은 피해호스트는 마킹된 패킷의 마크정보와 라우팅 맵을 이용하여 패킷이 경유한 라우팅 경로를 역추적할 수 있는 기법이다. 본 논문에서 제안하고 구현한 알고리즘을 침입 탐지시스템에 적용한다면 공격자의 해킹기법의 분석 및 강화된 보안정책을 설정할 수 있을 것이며, DDoS 공격으로 인한 시스템 및 네트워크의 서비스 거부로 발생할 수 있는 심각한 문제점들을 해결하여 한정된 네트워크와 시스템의 자원을 효율적으로 이용할 수 있을 것이다.

참 고 문 헌

[1] 포항공대 유닉스 보안 연구회, Security PLUS for UNIX, 영진출판사, 2001.

[2] S. Savage, D. Wetherall, A. Karlin, T. Anderson, "Practical Network Support for IP Traceback," In 2000 ACM SIGCOMM Conference, 2000.

[3] K.Park, H.Lee, "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack," Department of Computer Sciences, Purdue University, 2000.

[4] D.Song, A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," in IEEE INFOCOM 2001, 2001.
[5] M. Adler, "Tradeoffs in Probabilistic Packet Marking for IP Traceback," Department of Computer Science, University of Masachusetts, 2001.
[6] H. Krawczyk, M. Bellare, R. Canetti, "HMAC : Keyed-Hashing for Message Authentication," RFC 2104, 1997.
[7] 오창석, 데이터통신(수정판), 영한출판사, 2001.
[8] W. Stevens, TCP/IP Illustrated Volume 1, 2, Addison-Wesley, 1994.
[9] S. M. Bellovin, "Security Problems in the TCP/IP Protocol Suite," AT&T Bell Laboratories, Computer Communication Review, Vol. 19, No. 2, pp. 32-48, 1989.
[10] N. Nishio, N. Harashima, H. Tokuda, "Reflective Probabilistic Packet Marking Scheme for IP Traceback," IPSJSIG Notes system software and Operaing System Abstract, No. 090-009, 2002.

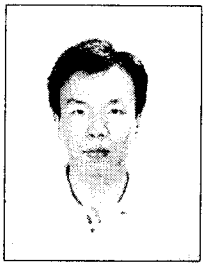
원 승 영(Seung-Young Won) 준회원

2002년 2월 : 충주대학교 컴퓨터공학과(공학사)
2002년 3월 ~ 현재 : 충북대학교 컴퓨터공학과(석사과정)
<관심분야> : 네트워크 보안, 뉴로 컴퓨터, 정보보호

한 승 완(Seung-Wan Han) 정회원

1994년 2월 : 전남대학교 전산학과 (이학사)
1996년 2월 : 전남대학교 전산통계학과 (이학석사)
2001년 8월 : 전남대학교 전산통계학과 (이학박사)
2001년 12월 ~ 현재 : 한국전자통신연구원 선임연구원
<관심분야> : 네트워크 보안, 암호화 이론, 계산 이론, 알고리즘

서 동 일(Dong-Il Seo) 정회원



1989년 2월 : 경북대학교 전자공학과 (공학사)
1994년 2월 : 포항공과대학교 정보통신공학과 (공학석사)
2000년 3월 ~ 현재 : 충북대학교 전자계산학과(박사과정)
1994년 ~ 현재 : 한국전자통신연구원 선임연구원
<관심분야> : 인터넷 정보보호, 컴퓨터 통신, 네트워크

김 선 영(Sun-Young Kim) 준회원

2001년 2월 : 한밭대학교 전자공학과 (공학사)
2003년 2월 : 충북대학교컴퓨터공학과 (공학석사)
2003년 3월 : 충북대학교컴퓨터공학과 (박사과정)
<관심분야> : 네트워크 보안, Embeded System, 정보 보호

오 창 석(Chang-Suk Oh) 종신회원

1978년 2월 : 연세대학교 전자공학과 (공학사)
1980년 2월 : 연세대학교 전자공학과 (공학석사)
1988년 8월 : 연세대학교 전자공학과 (공학박사)
1985년 ~ 현재 : 충북대학교 전기전자컴퓨터공학부 교수
1982년 ~ 1984년 : 한국전자통신연구원 연구원
1990년 ~ 1991년 : 미국 Stanford대학교 객원교수
<관심분야> : 컴퓨터 네트워크, 뉴로 컴퓨터, 정보 보호